

## **AMENDMENTS TO THE CLAIMS**

This listing of claims replaces all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) In a computing environment, a computer-implemented method for composing two-dimensional vector graphics, the method comprising:

receiving a function call for composing two-dimensional vector graphics via an application programming interface of a graphics processing environment, the function call comprising a native format including a markup language data, the markup language data comprising direct code calls, object model code calls, and XML-based markup;

a parser/translator, the parser/translator enabled to interpret each of direct code calls, object model code calls, and XML-based markup, interpreting the markup language data in its native format to cause data in a scene graph to be modified; and

causing a change in a graphics display in response to the modification of data in the scene graph.

2. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises causing initialization of a new instance of a visual class.

3. (Original) The method of claim 2 wherein causing data in the scene graph to be modified comprises invoking code to associate a transform with a visual object in the scene graph.

4. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking code to place a drawing visual into the scene graph.

5. (Original) The method of claim 4 further comprising, causing a drawing context to be returned, the drawing context providing a mechanism for rendering into the drawing visual.

6. (Original) The method of claim 2 wherein causing data in the scene graph to be modified comprises invoking code to associate brush data with a visual object in the scene graph.

7. (Original) The method of claim 6 wherein the brush data comprises receiving data corresponding to a solid color.

8. (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a linear gradient brush and a stop collection comprising at least one stop.

9. (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a radial gradient brush.

10. (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to an image.

11. (Original) The method of claim 10 further comprising, receiving markup corresponding to an image effect to apply to the image.

12. (Original) The method of claim 1 further comprising, receiving markup corresponding to pen data that defines an outline of a shape.

13. (Original) The method of claim 1 wherein the markup corresponds to a shape class comprising at least one of the set containing rectangle, polyline, polygon, path, line and ellipse shapes.

14. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a geometry-related function to represent a rectangle in the scene graph data structure.

15. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a geometry-related function to represent a path in the scene graph data structure.

16. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a geometry-related function to represent a line in the scene graph data structure.

17. (Original) The method of claim 1 wherein the markup is related to hit-testing a visual in the scene graph data structure.

18. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to transforming coordinates of a visual in the scene graph data structure.

19. (Original) The method of claim 1 wherein the markup is related to calculating a bounding box of a visual in the scene graph data structure.

20. (Previously Presented) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function via a common application programming interface to a visual object in the scene graph data structure.

21. (Original) The method of claim 1 further comprising invoking a visual manager to render a tree of at least one visual object to a rendering target.

22. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place a container object in the scene graph data structure, the container object configured to contain at least one visual object.

23. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place image data into the scene graph data structure.

24. (Original) The method of claim 23 wherein causing data in the scene graph to be modified comprises invoking a function to place an image effect object into the scene graph data structure that is associated with the image data.

25. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place data corresponding to text into the scene graph data structure.

26. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to provide a drawing context in response to the function call.

27. (Original) The method of claim 26 wherein the function call corresponds to a retained visual, and further comprising, calling back to have the drawing context of the retained visual returned to the scene graph data structure.

28. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place a three-dimensional visual into the scene graph data structure.

29. (Original) The method of claim 28 wherein causing data in the scene graph to be modified comprises mapping a two-dimensional surface onto the three dimensional visual.

30. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place animation data into the scene graph data structure.

31. (Original) The method of claim 30 further comprising communicating timeline information corresponding to the animation data to a composition engine.

32. (Original) The method of claim 31 wherein the composition engine interpolates graphics data based on the timeline to animate an output corresponding to an object in the scene graph data structure.

33. (Original) The method of claim 32 wherein the composition engine is at a low-level with respect to the scene graph.

34. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place an object corresponding to audio and/or video data into the scene graph data structure.

35. (Original) The method of claim 1 wherein causing data in the scene graph to be modified comprises changing a mutable value of an object in the scene graph data structure.

36-62 (Canceled).